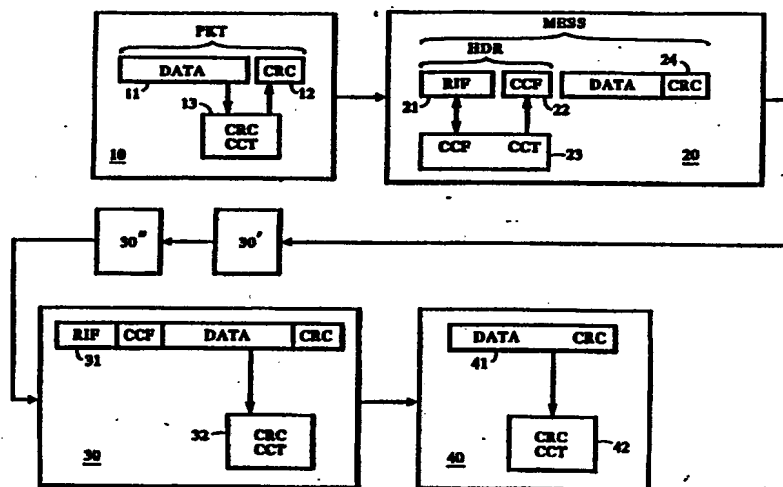




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : H04L 1/00	A1	(11) International Publication Number: WO 94/00937 (43) International Publication Date: 6 January 1994 (06.01.94)
(21) International Application Number: PCT/US93/05918 (22) International Filing Date: 21 June 1993 (21.06.93) (30) Priority data: 9213272.9 23 June 1992 (23.06.92) GB (71) Applicants: DIGITAL EQUIPMENT CORPORATION [US/US]; 146 Main Street, Maynard, MA 01754 (US). DIGITAL EQUIPMENT INTERNATIONAL LTD. [CH/CH]; 1, grand' Places, CH-1700 Fribourg (CH). (72) Inventors: HIGGINSON, Peter; Leslie ; 7 Tolmers Ave- nue, Cuffley, Herts EN6 4QE (GB). BERENT, Anthony, Neil ; 16 Raven Road, Hook, Hants RG27 9HH (GB).	(74) Agents: NATH, Rama, B. et al.; Digital Equipment Cor- poration, 111 Powdermill Road (MS02-3/CI), Maynard, MA 01754 (US). (81) Designated States: JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i>	

(54) Title: MESSAGE IN PACKET FORM WITH HEADER CONSTITUTED BY ROUTING INFORMATION AND A CRC CHECK SEQUENCE



(57) Abstract

A packet is originated in a unit (10) as a data field DATA (11) plus a CRC (cyclic redundancy check) check field CRC (12) by a CRC circuit (13). This packet has a header HDR (with a routing information field RIF) added to it in a unit (20), converting it into a message for transmission through a message network. A check correction field CCF is computed by unit (23) in unit (20), by looking up precomputed check subfields stored with the routing subfields (the routing information field being constructed by selecting from the stored subfields), such that the CRC field is a valid CRC check field for the complete message. At the destination, unit (30) can be the final user unit, checking the entire message and extracting the data field DATA therefrom; the DATA field does not need to be checked, as the CRC field acts as a check both for the data field DATA alone and the entire message. (Alternatively, the message can be checked by a final switching unit (30) using a standard CRC check circuit (32) (and similarly at intermediate units 30', 30'') and the original packet can be checked by another standard CRC check circuit (42) in the final user unit (40)).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NE	Niger
BE	Belgium	GN	Guinea	NL	Netherlands
BF	Burkina Faso	GR	Greece	NO	Norway
BG	Bulgaria	HU	Hungary	NZ	New Zealand
BJ	Benin	IE	Ireland	PL	Poland
BR	Brazil	IT	Italy	PT	Portugal
BY	Belarus	JP	Japan	RO	Romania
CA	Canada	KP	Democratic People's Republic of Korea	RU	Russian Federation
CF	Central African Republic	KR	Republic of Korea	SD	Sudan
CG	Congo	KZ	Kazakhstan	SE	Sweden
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovak Republic
CM	Cameroon	LU	Luxembourg	SN	Senegal
CN	China	LV	Latvia	TD	Chad
CS	Czechoslovakia	MC	Monaco	TC	Togo
CZ	Czech Republic	MG	Madagascar	UA	Ukraine
DE	Germany	ML	Mali	US	United States of America
DK	Denmark	MN	Mongolia	UZ	Uzbekistan
ES	Spain			VN	Viet Nam
FI	Finland				

MESSAGE IN PACKET FORM WITH HEADER CONSTITUTED BY ROUTING INFORMATION
AND A CRC CHECK SEQUENCE

Extended ECC System

5 The present invention relates to the provision of check quantities for digital data, particularly for data transmission.

10 The use of ECC (error detecting and/or correcting codes) for digital data is well established. ECC is used where the data is to be stored or transmitted by some means which is liable to occasional error. The data has a check portion or field appended to it to form a packet when it is entered into the storage or transmission means. When the
15 packet is retrieved, the check field is used to detect whether errors have occurred and, in some situations, to correct them. A new check field may for example be calculated from the data field of the packet and compared with the check field in the packet. It is convenient to
20 distinguish the data and check portions of the packet, but the check field checks the entire packet, ie itself as well as the data field.

25 A variety of ECC techniques have been proposed. Obviously, ease of implementation is important; for practical utilization, there must be some easily implemented algorithm for calculating the check field from the data field. If the system is to be used for error

correction, then there must also be an effective technique for calculating the required corrections to the data field from the (erroneous) packet. However, if only error detection is required, then there is no need for an efficient means of determining the precise nature of the error. Pure error detection, with no correction, is for example commonplace in many message transmission networks, because if an error is detected, the message can be retransmitted.

10

A widely used group of ECCs is cyclic ECCs. These are generally known as CRCs (cyclic redundancy check codes), and can readily be used for checking fields of variable length. A CRC code generates a check field which is termed the CRC field and is normally appended to the end of the data being checked. The present invention finds its main application with CRCs, and will be described in terms of CRCs rather than ECCs generally.

20

For present purposes, a CRC can be regarded as involving the division of the data field (regarded as a binary value) by a fixed binary value termed a polynomial, using arithmetic over the Galois field GF2 (ie modulo 2 and without carries between different positions). The remainder is the CRC or check field. The classical technique for calculating a CRC involves using a shift register with a feedback circuit including XOR gates at the positions corresponding to the 1s (powers of x) in the

25

polynomial. (Details of this can be found in ~~E~~Error
Correcting Codes\$, 2nd edition, Peterson & Weldon, MIT
1972.)

5 Many digital systems consist of a variety of
physically distributed units, and require data to be passed
between the various units. In some such systems, the
connections may be relatively simple; eg there may be
separate connections between each pair of units, or there
10 may be a single master unit to which all other units are
connected. In most large multi-unit systems, however, the
connections between the units, together with the interfaces
between the units and the connections, form a distinct
subsystem.

15

Such a subsystem is termed a digital data transmission
network. Such a network may consist of a switching
network having a number of switching nodes interconnected
with each other, or an area network such as a LAN (Local
20 Area Network), or a combination of switching networks and
area networks. (An area network has a common
communication medium with, normally, a large number of
units connected to it.)

25 The present invention is mainly, though not
exclusively, concerned with such digital data transmission
networks.

With such networks, it is the responsibility of the network to achieve the proper carriage of data between the end units connected to the network. This typically involves determining suitable routing for the data (eg
5 through a suitable sequence of switching nodes). The routing information required for this is a function of the network, independent of the end units. The switching network therefore normally has to add a header including routing information to the data as received from the source
10 (originating) end unit. This header is either stripped off just before the message is delivered to the destination (receiving) end unit, or is ignored by that unit.

This complicates the application of ECC.

15

The original data, as generated by the originating end unit, will often be generated with its own CRC, and for various reasons, it is often highly desirable for the data to retain this CRC throughout its journey to the receiving
20 end unit. The transmission network must therefore treat this combination of data plus its CRC as a unit (which we will term a packet). The transmission network adds a header to this packet (we will term this combination of header plus packet a message). (It will be realized that
25 terms such as packet and message are used here with specific senses which may not be identical with their usual meanings in the context of message transmission networks.)

Just as for the data generally, it is often desirable for the header to be checkable; ie for ECC to be applied.

But the precise manner in which ECC (which means, in practice, CRC) is applied to the message involves various considerations and is not entirely straightforward.

One option is not to use a CRC for the header, in the expectation that an error in the header will result in the message failing to reach an end unit or, if it reached the wrong end unit, that end unit rejecting it; the sending end unit will then not receive an acknowledgement of its receipt, and will in due course resend it. This may not be acceptable.

A second option is to generate a header CRC for the header, so that the message consists of the header with its CRC, plus the packet (the data with its CRC).

A third option is for the packet to be encapsulated, with the header being added to the front of the packet and a combined CRC for the header plus packet being calculated and added at the end of the message.

A factor which has to be taken into consideration in selecting an option is whether the transmission network protocol is predetermined. If it is not, then any option may be chosen. But if it is, then the choice of option may be forced. The standard protocols require that the

message shall have a CRC at the end which covers the entire message, which forces the adoption of the third option.

The main object of the present invention is to provide
5 an improved technique for providing error detection for messages in message networks.

The crux of the present invention lies in including,
in the header of a message, a check correction field (CCF)
10 which is chosen such that the CRC of the packet is also a valid CRC for the entire message. This means that the message will automatically conform to the usual message network protocol. (It will be realized that the nature of the contents of the header is not relevant, although we
15 have described the header as containing routing information for use in a message network.)

The main advantage of this is that the CRC of the message forms a check for both the data of the original
20 packet and the header, so a single CRC check verifies the accuracy of both the message as a whole and the data alone.

This is in contrast to the known techniques discussed above. In the first of those, the header is not checked, and in the second, it is obvious that the header and the
25 data are checked independently. In the third, the packet encapsulation technique, it is possible for an error to occur in the packet as the CRC for the message (packet plus header) is being calculated; so a CRC check on the entire

message, using the CRC of the message, therefore verifies only the message header, and a second CRC check is therefore required on the packet extracted from the message to verify the integrity of the data.

5

The construction of the CCF will normally be performed in the interfacing between the originating end unit and the message network. Although the general functionality of this interfacing may be determined by the predetermined characteristics of the message network, it will normally be controllable by the user to the extent that it can be arranged to generate the CCF.

In some complex message networks, it may be necessary for the header to be changed as the message passes through the network. Again, the network nodes where this takes place will normally be controllable by the user to the extent that they can be arranged to generate a CCF for the new header which maintains the CRC of the packet as valid for the whole message.

An essential feature of the present system is that the CCF can be determined without reference to the packet, ie from the header alone. It will be realized that this places a certain constraint on the nature of the ECC used.

Broadly, the ECC must be linearly superposable (or be divisible into components which are linearly superposable) rather than being one which performs an encryption (of the

kind intended to discourage or prevent unauthorized reading of the information).

Broadly, the process of generating the check field must be linear, in the sense that the sum of the check fields for two different fields (the original data and the header) is the check field for the concatenation of the fields. This constraint is in fact satisfied by most popular ECCs, including in particular CRCs, and also by other codes with the appropriate structure of linear components, such as BCH codes.

The simple or basic CRC has been developed into a form known as "modified CRC". This involves the use of a second polynomial which is dependent on the length of the message. This form prevents two concatenated messages with CRCs from forming a single message with a correct CRC.

For example, the X.25 standard uses a polynomial

$$x^{15} + x^{14} + \dots + x^1 + 1)(x^{k+16} + 1),$$

where the first term in the product is a conventional CRC polynomial (with selected powers of x) and the second term is dependent on the length k of the data field. Although this is not linear, it can be separated into components which behave linearly for present purposes.

This modified CRC has now displaced the standard CRC virtually completely in message networks, and from here on we will work exclusively in terms of this modified CRC.

The calculation of the CCF depends on its location in the header. If it is at the end of the header, the calculation consists of calculating the CRC for the header (eg by using a conventional feedback shift register) and
5 then making a fixed adjustment to it, which can be treated as an inversion (ie XORing with an all 1s subfield).

It may be desirable to locate the CCF inside the header, ie not at the end. If this is the case, the
10 calculation of the CRC for the header requires a minor adjustment. The portion of the header following the CCF has to be divided not by the CRC polynomial but by its reciprocal polynomial, obtained by reading it in the reverse direction (eg the reciprocal polynomial of $x^3 + x^2 + x^0$ is $x^3 + x^1 + x^0$). The result of this calculation has
15 to be combined with the result for the portion of the header preceding the CCF by exclusive-ORing (without inversion). If the CCF is the last field of the header, then the result of the backwards calculation with the
20 reciprocal polynomial is a constant which depends only on the polynomial, and can easily be combined into the CCF.

The routing header will normally consist of a number of subfields, eg destination, source, and message type.
25 In many situations, most or all of these subfields will be known in advance. This means that their contributions to the CCF (check subfields) can be calculated in advance. The CCF can then generally be determined by a relatively

simple calculation based on those pre-calculated check subfields (XORing the check subfields together, and incorporating the fixed adjustment described above). Of course, if a new subfield is required, then its check
5 subfield has to be calculated.

If the subfields occur largely in fixed or mainly fixed combinations, the combined contribution of these subfields to the CCF can also, of course, be pre-calculated
10 as a single value. Similarly, it may be possible to include the result of the backwards calculation with the reciprocal polynomial (if required) in this value. Calculating the CCF will then involve combining the contribution of the fixed combination of subfields with the
15 check subfields of any further header subfields.

The precalculation of the value of a check subfield may be done by calculating it as a normal CRC check field for the header using a feedback shift register (taking all
20 the other header subfields as empty). This check subfield value must then be stored with the associated header subfield.

If the message is to pass through two different levels
25 of switching network, a header with a CCF field may be added at one stage and a second header with its own CCF field added at a later stage, with the original header (with its CCF field) being treated as part of the packet or

data. Such a switching network may also require the position of some components of the message to be changed.

Such components must also be treated as parts of the new header.

5

An embodiment of the invention will now be described, by way of example, with reference to the drawings, in which:

10 Fig. 1 is a block diagram of part of a a digital data transmission network; and

Fig. 2 is a more detailed block diagram of the CCF unit 23 in unit 20 of that network.

15

Referring now to the Fig. 1, a message transmission network is shown as including a source end unit 10 and three routing devices 20, 30, and 40. These illustrate part of the path of a particular data field through the
20 network, which will of course include many such units.

The source end unit 10 includes a register 11 in which the data field, which is to be sent to the destination unit 40, is generated. This unit 10 also includes a CRC
25 circuit 13 which is fed with data field and calculates a CRC therefrom, and a CRC register 12 into which the CRC so calculated is fed. The CRC circuit 13 may be a look-up table unit or a feedback shift register unit. The CRC

field is concatenated with the data field to form a packet PKT', which is transmitted by the unit 10.

The packet is passed to a routing unit 20 which
5 derives, from information provided by unit 10, the source and destination addresses (in the routing network) for transmitting the packet to a destination unit 30. In this routing unit, these addresses are combined (possibly after conversion to other forms) to form a route information
10 field RIF which is stored in a register 21. This unit 20 also includes a CCF circuit 23 which is fed with RIF field and calculates a CCF therefrom, and a CCF register 22 into which the CCF so calculated is fed. (The RIF field may be generated in register 21 and passed to the CCF circuit 23
15 for the calculation of the CCF, or it may be generated in the CCF circuit 23 as described below and passed to the RIF register 21.)

The RIF field in register 21 is concatenated with the
20 CCF field in register 22 to produce a header HDR. The packet itself from unit 10 is fed into a packet register 24, and this is concatenated with the header to produce a message MESS, which is transmitted by the unit 20.

25 Fig. 2 shows the CCF circuit 23 in more detail. This circuit comprises a set of look-up tables 25-1 to 25-3, one for each subfield of the RIF, and an XOR accumulator 26. Each of the look-up tables contains a set of values for its

subfield, together with the corresponding check subfield values. (It is convenient to store all three tables in a single memory, as shown, with all three subfields in each location, with each subfield value being accompanied by two
5 unused and empty subfields, shown shaded.) A control unit 27 selects, from each look-up table in turn, the desired subfield value, which is passed to the RIF register 21. The selection of the subfield values may be determined, for example, in accordance with the desired
10 final end unit to which the data is to be sent in combination with stored routing tables maintained by the unit 20 as part of the message network.

At the same time as the RIF subfield value is looked
15 up, its associated check subfield value is also looked up and passed to the accumulator 26, which adds the check subfield values according to polynomial arithmetic (bitwise XORing). A final fixed adjustment has to be made (as discussed above) to adjust the result for the fact that the
20 check system is a modified CRC system. This adjustment is in fact achieved by adjusting the check subfield values in any convenient one of the three look-up tables.

The final contents of the accumulator 26 are passed to
25 the CCF register 22. The CCF field generated by the CCF circuit 23 is thus concatenated with the CCF field to produce the header HDR as described above.

The CCF circuit 23 also includes a check subfield calculation unit 28 for calculating the check subfields for newly generated RIF subfields. (Obviously the calculation of new check subfields may impose some delay on the operation of the unit 20.)

Returning to Fig. 1, the message is sent by routing unit 20 to a second routing unit 30, which checks the message. The message has its RIF, CCF, DATA, and CRC fields entered into a register 31. The routing unit also includes a CRC circuit 32 which is fed with contents of register 31 and calculates a CRC from the entire contents of the register 31. The fact that the CRC is included in the calculation means that the result should be a predetermined value. The CRC circuit 32 emits an error signal if some other value results.

The CRC circuit 32 can be identical to the CRC circuit 13 of unit 10, apart from the fact that the CRC circuit 32 has to calculate its CRC from a longer quantity than the circuit 13.

We will assume that unit 30 is the final unit in the switching network. The header is stripped off the message by unit 30, which therefore passes only the packet part of the message on to the final destination unit, unit 40. This unit checks the packet, the whole of which (DATA and CRC fields) is entered into a register 41. The routing

unit also includes a CRC circuit 42 which is fed with the contents of register 41 and calculates a CRC therefrom. This should yield a predetermined value; the CRC circuit 42 emits an error signal if some other value results.

5

Obviously the message may be transmitted through any number of units similar to unit 30, as shown at 30' and 30".

10 Instead of unit 30 passing the packet to the final unit 40, unit 30 may itself be the final unit. In this case, the unit 30 will be as shown, but will extract the DATA field directly from register 31, discarding the RIF and CCF fields. The single check performed by the CRC
15 circuit 32 will in this case verify the integrity of both that the DATA field and the header.

A further possibility is that the header may need to be changed at some stage along the route of the message.

20 For this, a unit combining the features of units 20 and 30 would be required. More specifically, this would involve adding a CRC circuit (like unit 32) to unit 20. The incoming message would be entered into the complete message register, and its CRC checked by the CRC circuit. The
25 incoming header would then be discarded, the new RIF field would then be determined, and finally the new CCF field would be determined by the unit 23.

The CRC circuit 42 can be identical to the CRC circuits 13 and 32 of units 10 and 30, apart from the fact that the CRC circuit 42 has to calculate its CRC from a quantity of different length to the quantities dealt with
5 by the circuit 32. Further, the CRC fields which are used to check the message in unit 30 and the packet in unit 40 are identical.

The unit 40 may be the destination end unit for the
10 data field; alternatively, the packet may be forwarded further through the network, eg with a new RIF being added by a unit corresponding to unit 20.

In summary, in the present system a packet is
15 originated in a unit 10 as a data field DATA 11 plus a CRC (cyclic redundancy check) check field CRC 12 by a CRC circuit 13. This packet has a header HDR (with a routing information field RIF) added to it in a unit 20, converting it into a message for transmission through a message
20 network. A check correction field CCF is computed by unit 23 in unit 20, by looking up precomputed check subfields stored with the routing subfields (the routing information field being constructed by selecting from the stored subfields), such that the CRC field is a valid CRC check
25 field for the complete message. At the destination, unit 30 can be the final user unit, checking the entire message and extracting the data field DATA therefrom; the DATA field does not need to be checked, as the CRC field acts as

a check both for the data field DATA alone and the entire message. (Alternatively, the message can be checked by a final switching unit 30 using a standard CRC check circuit 32 (and similarly at intermediate units 30', 30'') and the
5 original packet can be checked by another standard CRC check circuit 42 in the final user unit 40.)

Claims

- 5 1 In a message transmission system, check generating means for adding, to a packet which has a check field, a header routing information field (RIF) to form a message, characterized by means (23) for calculating, from the header information field, a check correction field (CCF) which is incorporated in the header (HDR) and preserves said check field as valid for the entire message (MESS).
- 2 Check generating means according to claim 1, characterized in that the check field is a cyclic
15 redundancy check (CRC) field.
- 3 Check generating means according to claim 2, characterized in that the check field is a modified cyclic redundancy check field utilizing a CRC polynomial and a
20 second polynomial dependent on the length of the packet.
- 4 Check generating means according to either of claims 2 and 3, characterized by means (28) for calculating the check subfields for header subfields, including means for
25 dividing by the CRC polynomial.
- 5 Check generating means according to any of claims 2 to 4, characterized by means (28) for calculating the check

subfields for header information subfields, including means for dividing by the reciprocal of the CRC polynomial.

6 Check generating means according to any previous
5 claim, characterized by look-up table storage means (25-1
to 25-3) for storing values of subfields of the header
information field (RIF) together with associated check
subfields, and means (26) for combining, for subfield
values selected for inclusion of the header information
10 field, the associated check subfield values to form the
check correction field (CCF).

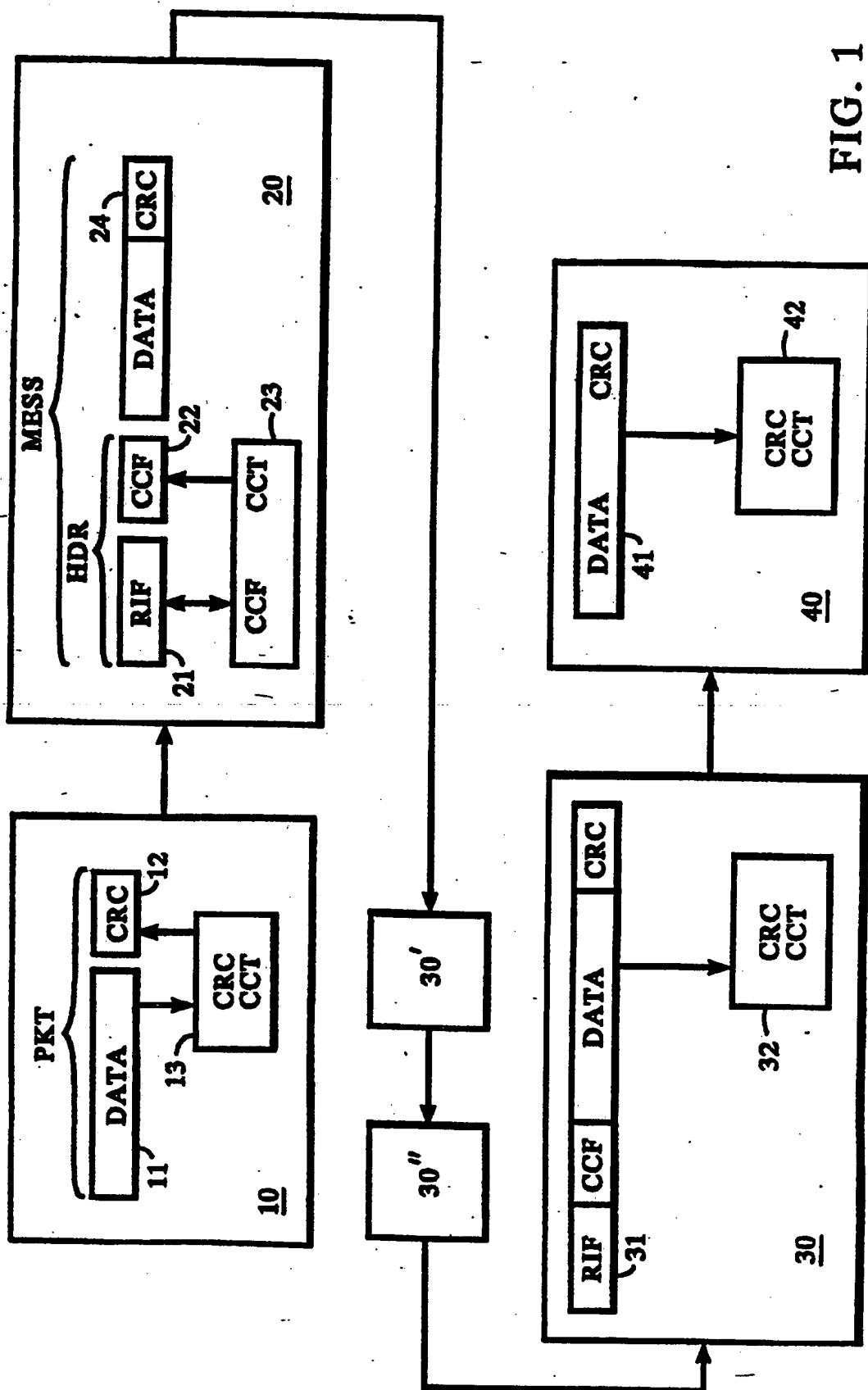


FIG. 1

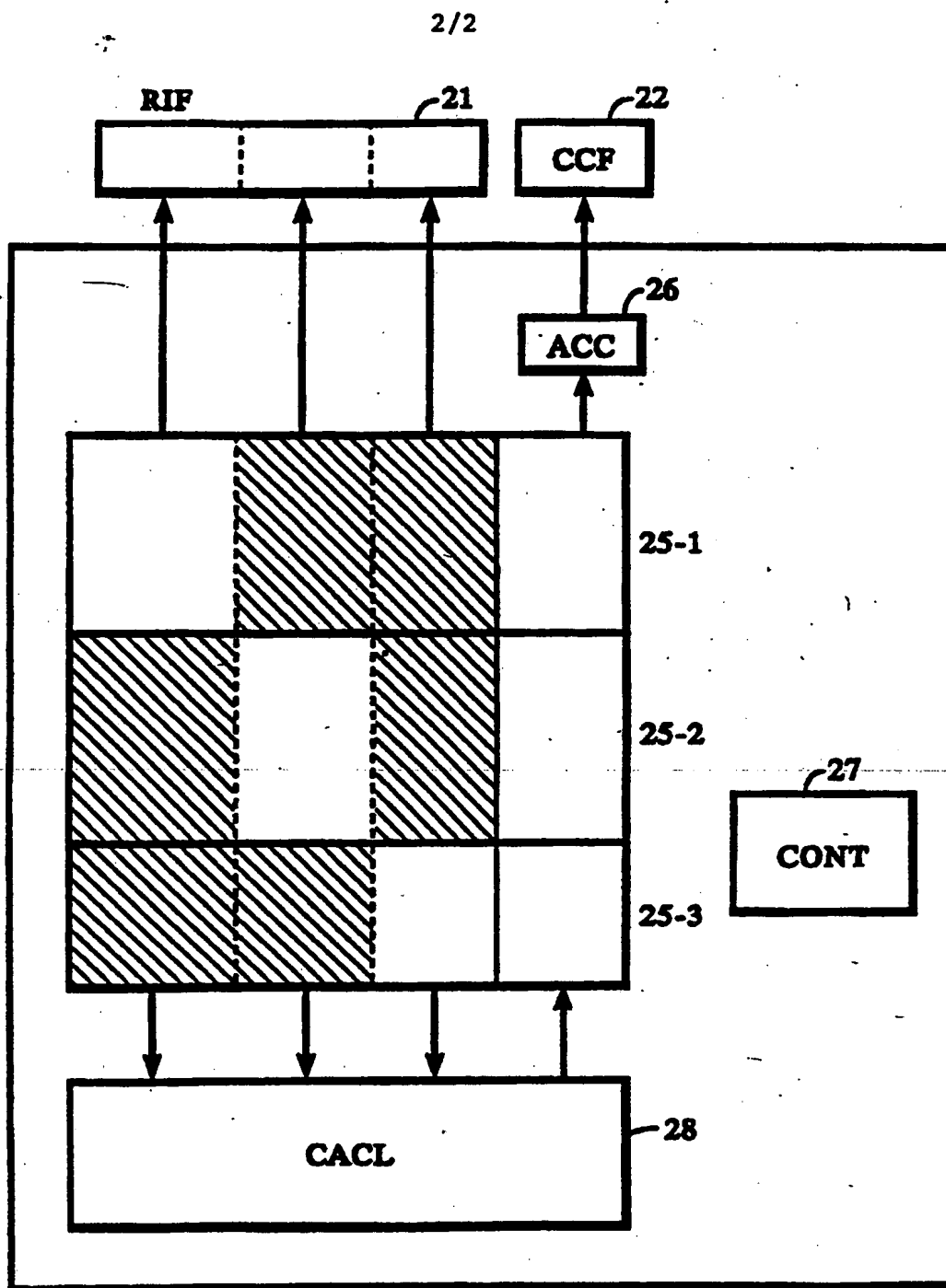


FIG. 2